Report No. 72-0012
Contract No. NAS8-27359

MULTI-PROCESSING CONTROL SYSTEM FOR THE SEL 840MP (MPCS/1)

USERS GUIDE

VOLUME II - OPERATIONS GUIDE

March 27, 1972

Prepared for:

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
George C. Marshall Space Flight Center
Marshall Space Flight Center, Alabama 35812

M&S COMPUTING, INC.

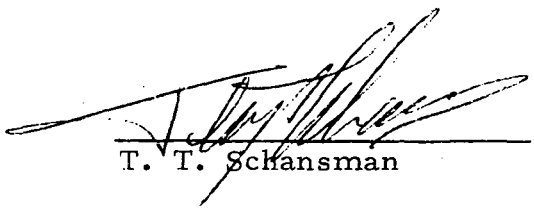
PREFACE

This document provides the information required for the generation and operational use of the SEL 840MP Multi-Processing Control System - Version I (MPCS/1). This system was developed by M&S Computing under Contract No. NAS8-27359 for NASA/MSFC.

Participating personnel were:

W. F. Anderson
J. R. Conway
L. C. Keller
M. L. Williams

Approved by:



T. T. Schansman

TABLE OF CONTENTS

Section			Page
List of Figures			iv
1.	INTRODUCTION		1
2.	SYSTEM INITIALIZATION		7
	2.1	Initial System Load Procedure - 840MP	7
	2.2	System Reload Procedure	8
	2.3	Initial System Load Procedure - 810B	9
3.	JOB TASK TABLE GENERATION		11
	3.1	Compile-Load-and-Go	11
	3.2	Load-and-Go	11
4.	MPCS COMMAND LANGUAGE		15
	4.1	Command Language Operators	18
		4.1.1 Process Command (PC)	18
		4.1.2 Omit Command (OC)	20
		4.1.3 Device Change (DC)	20
		4.1.4 Dump Program (DP)	20
		4.1.5 Call Program (CP)	21
		4.1.6 Find Program (FP)	21
		4.1.7 Delete Name (DN)	21
		4.1.8 List Program (LP)	21
		4.1.9 Octal Patch (OP)	22
		4.1.10 Assembler Call (AS)	22

TABLE OF CONTENTS
(continued)

Section			Page
	4.1.11	FORTRAN Call (FC)	24
	4.1.12	*Relocating Loader Call (RL)*	25
	4.1.13	Dump Absolute (DA)	27
	4.1.14	Octal Dump (OD)	28
	4.1.15	Execute Program (EX)	28
	4.1.16	Source Update (UD)	28
	4.1.17	File Assignment (FA)	30
	4.1.18	Interrupt Subroutine Names (IN)	31
	4.1.19	Restore Standard Assignments (RS)	31
	4.1.20	Copy Library (CL)	31
	4.1.21	Position Peripheral (PP)	32
	4.1.22	Backspace File (BF)	32
	4.1.23	Search Magnetic Tape Record (SR)	33
	4.1.24	Search Magnetic Tape File (SF)	33
	4.1.25	Backspace Magnetic Tape Record (BR)	33
	4.1.26	Advance Magnetic Tape Record (AR)	33
	4.1.27	Advance Magnetic Tape File (AF)	34
	4.1.28	Select Peripheral Switch (SS)	34
	4.1.29	Generate Job Task Table (JT)	34
4.2	Display Console Functions		34
	4.2.2	Light Pen Input	37

TABLE OF CONTENTS
(continued)

Section			Page
		4.2.3 Function Switch Input	38
5.	DISPLAY LIBRARY GENERATION		39
	5.1	Process DLL Card Images	39
	5.2	810B Disk Library Generation	41
6.	SYSTEM ERROR SUMMARY		43
	6.1	Error Conditions Causing User Abort	43
	6.2	I/O Error Conditions for General I/O Devices	44
	6.3	Display Message Summary	44
	6.4	I/O Error for Hybrid Services	47
7.	SYSTEM GENERATION		49
	7.1	Bootstrap Deck Creation	49
	7.2	System Tape Creation	50
	7.3	System Disk Creation	50
		7.3.1 Complete Initialization Procedure	50
		7.3.2 Partial Initialization Procedure	53
	7.4	810B System Generation	54

LIST OF FIGURES

No.	Title	Page
1-1	840MP Hardware Configuration	2
1-2	SEL Display System	3
1-3	Processor Domains	4
1-4	Initial BAR Settings for Processor Domains	5
3-1	Sample Deck Setup	12
4-1	Standard Device LDN and Mnemonic Assignment	17
4-2	Command Language Operator Summary	19
4-3	Display Console Character Set	36
5-1	Display Librarian Input Deck Setup	40
6-1	I/O Error Conditions	45
7-1	Title Card Format	51
7-2	System Generation Deck Setup	52

1. INTRODUCTION

A detailed description of the SEL 840MP can be found in "Reference Manual, SEL 840MP General Purpose Computer System", form No. 301-095098-001, System Engineering Laboratories. The functional systems configuration of the 840MP is depicted in Figure 1-1. Figure 1-2 depicts the hardware configuration of the 810B display system.

Each processing element has access to its own private memory, as well as all of shared memory. Simultaneous access of a shared memory module by the processors, is resolved by assigning fixed priorities to the processors. This particular access resolution scheme may result in the dominance of a particular processor, making true simultaneous execution from a shared memory module impractical.

Input/Output devices may be attached to a processing element channel or to shared memory. When they are attached to a processing element channel, they can access a particular private memory and all of shared memory, but can only be commanded by the particular processing element. When the devices are attached to a shared memory channel, they can only access shared memory, but may be commanded by any processor.

For MPCS/1, the three private memories and all of shared memory have been partitioned into three separate processor domains. These domains insure that no conflicts for core requirements will occur for up to three separate jobs resident in the three processors at the same time. Each user configured domain provides 32K, where K equals 1024 locations. The first 8K is required by MPCS/1. This enables a particular application program to reside in or access 24K without a change in Bank Address Registers (BAR). Except for the memory that is dedicated to MPCS/1, application programs may access all of private and all of shared memory simply by changing the BARs. MPCS/1 does not resolve resource conflicts. Application programmers must insure that, when a BAR change is made, the new core configuration does not conflict with jobs currently resident in other processors. Figure 1-3 depicts the three processor domains and Figure 1-4 provides the initial BAR setting for each domain. It should be noted that BAR 0 is always dedicated for the exclusive use of MPCS/1.

840MP HARDWARE CONFIGURATION

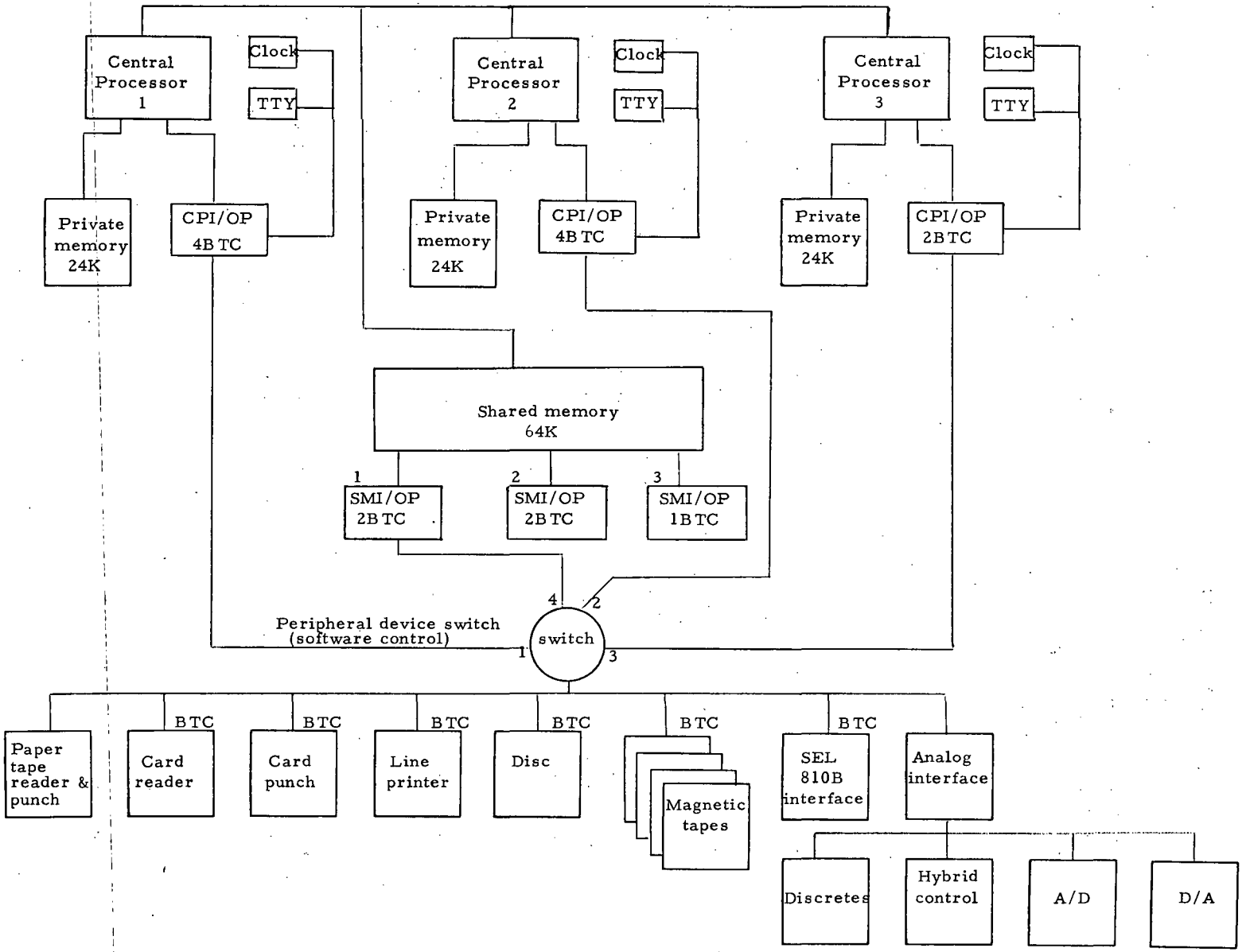


Figure 1-1

SEL DISPLAY SYSTEM

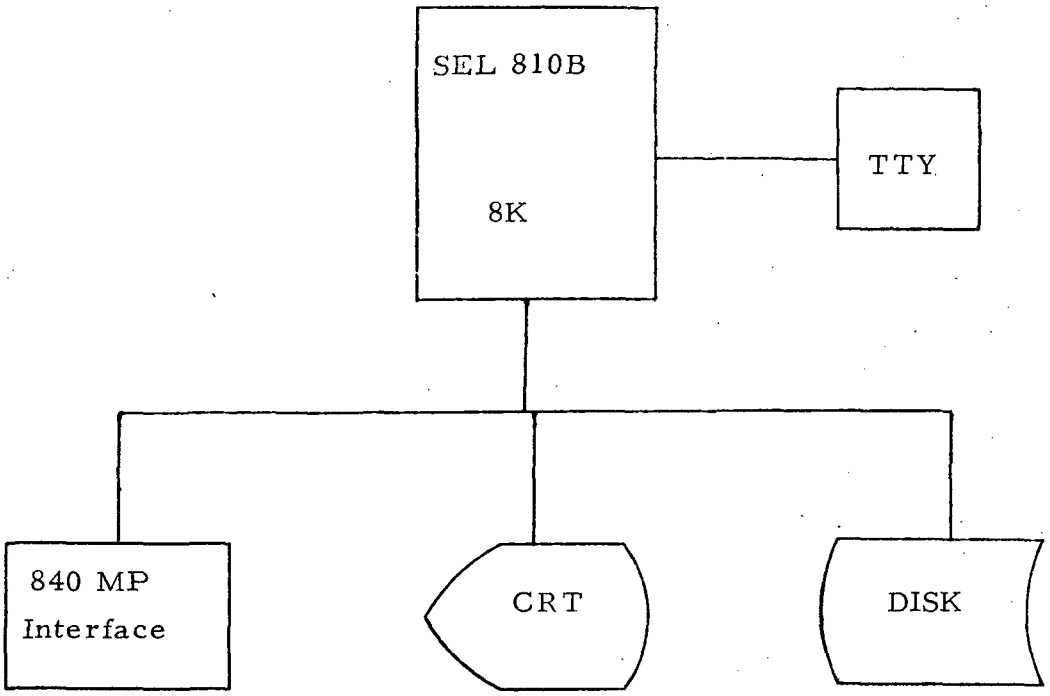


Figure 1-2

PROCESSOR DOMAINS

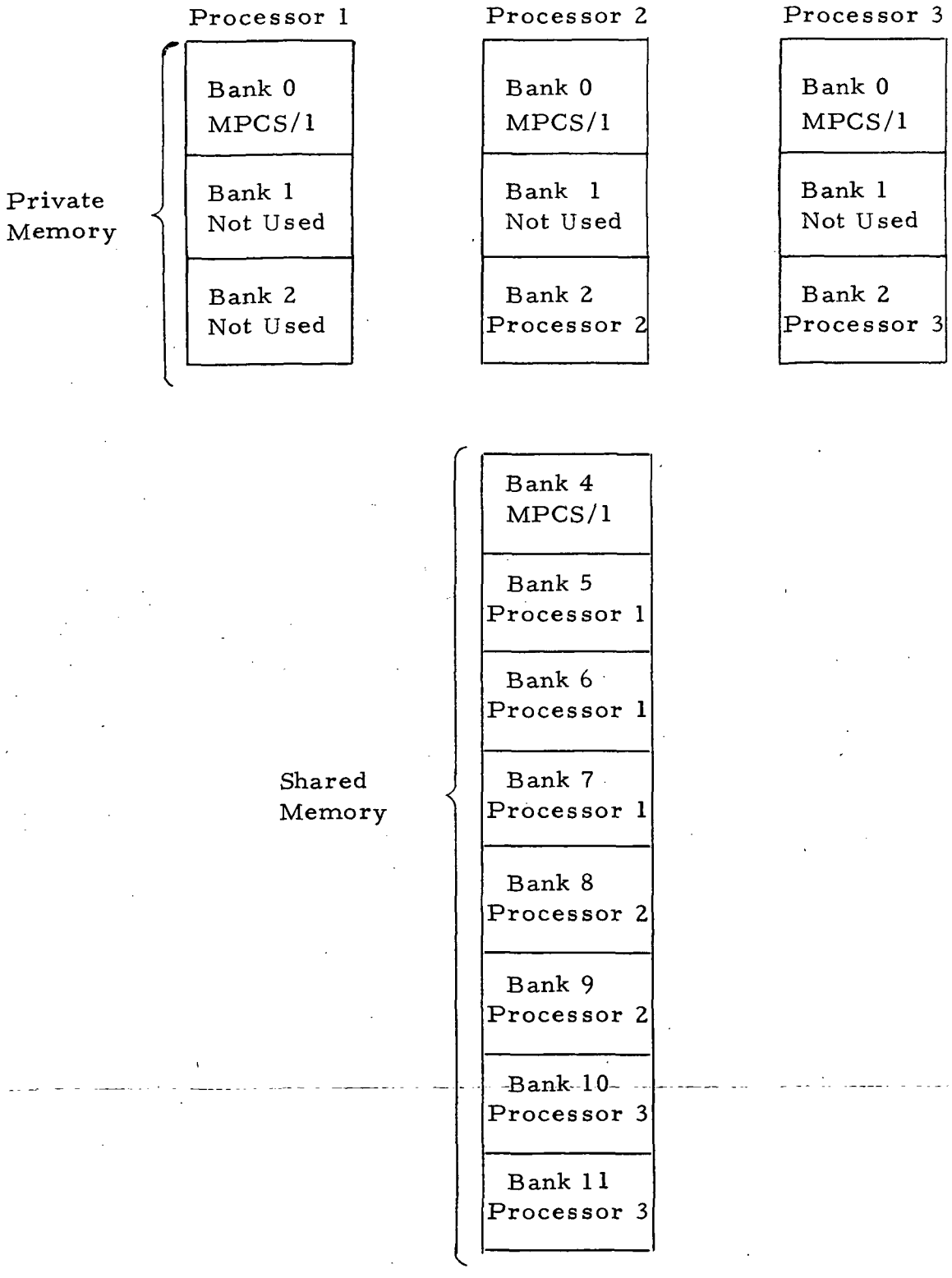


Figure 1-3

INITIAL BAR SETTINGS FOR PROCESSOR DOMAINS

DOMAIN	BAR 0	BAR 1	BAR 2	BAR 3
MPCS/1	0	4	1	2
Processor 1	0	5	6	7
Processor 2	0	8	9	2
Processor 3	0	10	11	2

Figure 1-4

2. SYSTEM INITIALIZATION

This section will describe the procedure used to initially load one or more central processors and will also describe the procedure used to reload one central processor while others are in operation. The system uses 8K of private memory (00000 - 17777) and 8K of shared memory (20000 - 37777). Private memories are loaded one at a time through the associated processor. Shared memory can be initialized through any processor, and only has to be initialized once.

2.1 Initial System Load Procedure - 840MP

The system is initially loaded as follows:

(1) Prepare the desired processor for loading:

(a) Disable memory protect switch of that processor.

(b) Place peripheral switch in the position corresponding to that processor.

Processor number 1 is the center computer, processor number 2 is the one on the right, and processor number 3 is the one on the left as you face the three computers.

(2) Load the MPCS bootstrap card via the load switch on the control console.

Input and execution of the program contained on this card will cause the private memory of the central processor to be loaded and will transfer control to MPCS.

(3) As MPCS is entered, the following message is output:

```
SWITCH REQUIREMENTS
  C = CP I/OP
  S = SM I/OP
  R = SYSTEM READY
```

The C response will cause shared memory to be initialized for use with CP I/OP (I/O attached to processing element channel). The S response will cause shared memory to be initialized for use with SM I/OP (I/O attached to shared memory channel). The R response advances the load procedure. This should only be used if shared memory was previously initialized. If an

illegal character is detected as the response, the initial message above will be output again and the typewriter will be enabled for another response.

(4) After a correct response is made, the following message is output:

ENABLE PROGRAM PROTECT

MPCS will wait until the program protect switch on the control console is enabled before it will continue.

(5) When the protect switch is enabled, "MPCS/1" will be typed and the load procedure for this computer is completed.

(6) Steps 1 through 5 have to be repeated for each processor to be used.

(7) If all desired processors have been initialized, the peripheral switch should be moved to R (Remote). If this last step is omitted, the message: "INSURE SWITCH IS IN REMOTE POSITION", will be typed the first time a job is requested from one of the central processors. When the message is typed, MPCS will wait until the switch is in the R (Remote) position before it will continue.

(8) If the 810B Display is to be used, the "Initial System Load Procedure - 810B" (reference Paragraph 2.3) now has to be performed.

2.2 System Reload Procedure

If it is necessary to reload a central processor or to load a processor that was not initially loaded, the following procedures should be followed:

(1) Insure that all jobs in all central processors have completed execution.

(2) Move the peripheral switch to the position that corresponds to the number of the processor that is desired to be loaded and follow the procedure described under "Initial System Load".

2.3 Initial System Load Procedure - 810B

Two components of the MPCS display support software, the Display Library Loader and the Display Processor, execute in the SEL 810B. Initialization of the 810B should be performed prior to the execution of any task in the 840MP which requires display services. The 810B is initialized in the following manner:

(1) Hand load the following bootstrap:

Location	Contents
0001	130113
0002	000010
0003	130113
0004	100001
0005	110640
1060	000640
1061	100140

(2) Depress Master Clear

(3) Depress Single Cycle three times

(4) Depress Start

At this point the SEL 810A/810B Operating System (OS) has been loaded and is ready to accept input commands from the 810B console keyboard.

The above steps are required only when the system bootstrap is not in the computer. Under normal operating conditions, the system can be reinitialized at any time by loading the address 640_8 into the Program Counter and depressing Start twice.

(5) Either the Display Library Loader or the Display Processor can now be loaded and executed.

Ordinarily, the user will require only the Display Processor which can be invoked by entering the following commands via the keyboard:

```
/POSF,BI, MPCSDP.
/XFER, BI, F=0, L=5000.
/GOTO, R=300.
```

The Display Processor will display a "Ready" message on the CRT and will wait for the initial display request from an application task executing in the 840MP.

If the Display Processor has been previously loaded into the computer, it can be activated by simply loading 300_8 into the Program Counter and depressing Start twice. This procedure is most convenient when it is desired to restart a display application. However, if it cannot be restarted using this technique, it will be necessary to reinitialize the operating system and the Display Processor as discussed in the preceding paragraphs.

The Display Processor does require the existence of a library of preformatted displays in disk storage. Procedures for generating and loading this library (Display Library Loader) are described in Section 5.

3. JOB TASK TABLE GENERATION

Jobs which utilize MPCS task control facilities, as discussed in Volume I, Section 2, are required to have a Job Task Table (JTT) defining the task structure for the job. The JTT consists of a number of subtables called Task Queue Items (TQI) which contain information describing each task. A TQI is required for each task in the job.

The JTT is generated from input control cards as discussed in Sections 2.3 and 2.4 of Volume I. The control cards are processed to produce object code, similar to that produced by the Assembler and FORTRAN Compiler, which can be interpreted by the Relocatable Loader. When the job is loaded for execution, the TQI's within the JTT are linked to their associated tasks and the JTT is then used as the central task directory for all task control functions.

Either of two techniques can be used to produce the JTT object code depending on the manner desired for loading the job for execution. These techniques are discussed in the following sub-sections. Note that in either case the JTT is considered to be the MAIN program for the job and, as such, must be loaded before the individual tasks.

3.1 Compile-Load-and-Go

When it is desired to process a job following compilation or assembly of task source code, the following operator commands can be used (refer to Figure 3-1 for a typical deck setup):

```
JT
FC
RL
PC
```

The first command causes the JTT processor to be loaded and executed. It will read the JTT input control cards and produce JTT object code for the Relocatable Loader. The second command will invoke the FORTRAN Compiler for compilation of task source code and the third command will cause the Relocatable Loader to process the generated object code. Following execution of the Relocatable Loader, the first task of the job will be scheduled automatically.

3.2 Load-and-Go

If a job is to be executed directly from object code, the JTT input control cards can be preprocessed to produce a JTT object deck.

SAMPLE DECK SETUP

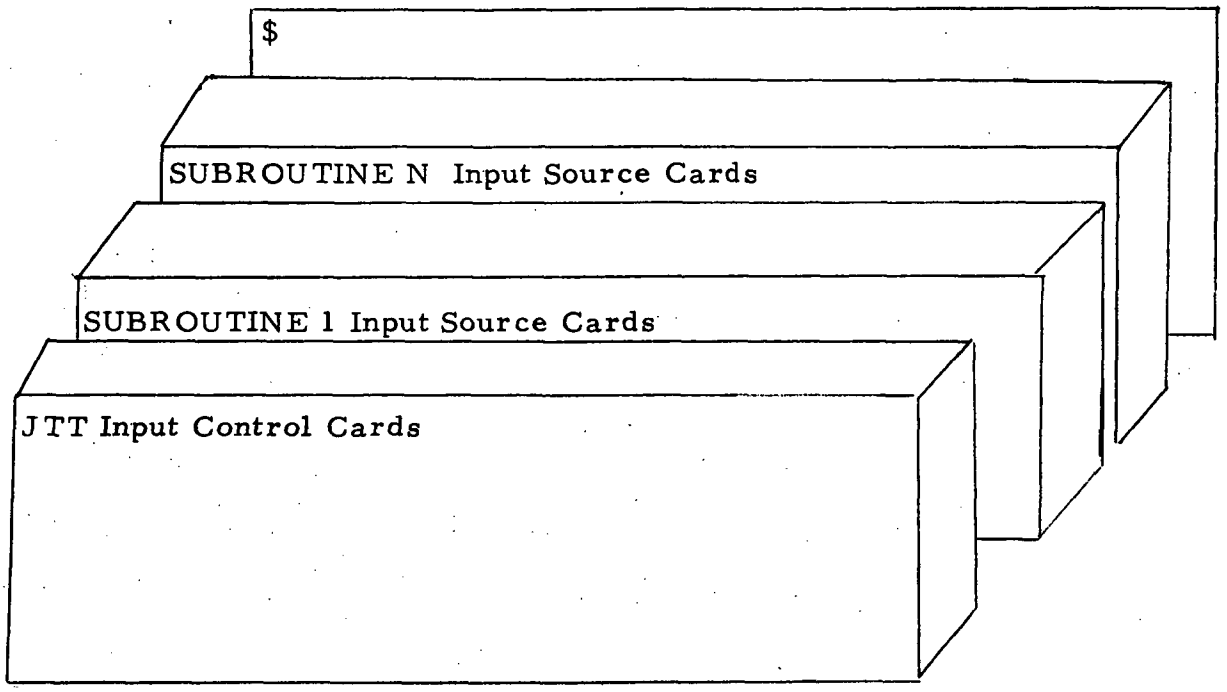


Figure 3-1

The following operator commands can be used to perform this function:

```
FA BO, CP
JT
PC
```

These commands will cause the input JTT control cards to be read and processed with the resulting object code punched out on cards. JTT object cards can then be used as an object deck for a MAIN program would normally be used.

4. MPCS COMMAND LANGUAGE

The command language for MPCS consists of a group of operators, each of which specifies some particular function or service provided by MPCS. By logically sequencing these command language operators, MPCS users can direct and control all of the system resources including the software language processors, loaders, and utility programs, as well as diagnostics and general service routines. Additionally, users may add application jobs to the system bulk storage device from where they may subsequently be loaded and executed. MPCS permits growth or change of system capability by providing command language operators to maintain and control the extensive subroutine library.

MPCS users familiar with EXEC and CHANE will have little difficulty in becoming familiar with the mnemonics and services of each of the MPCS command language operators. Mnemonics of the operators are identical to those found in EXEC; syntax and services are identical to those found in CHANE. Additional operators have been provided to enable users to structure application tasks for the MPCS environment and to control the 840MP peripheral switch.

In its quiescent state, MPCS waits for command language operators to be input via the command device which is initially and generally the console teletype keyboard. When using the teletype keyboard as the input media for command language operators, each entity is terminated with a carriage return. Any other device designated as the command device must receive card image (80 character) entities.

The general format of the MPCS command language operator is as follows:

MM Parameter 1, Parameter 2... Parameter N

or:

MM

These formats illustrate that the command language operators consist of unique two character mnemonics (MM), followed by a single space, followed by a list of parameters where parameters are required. Each parameter except the last is followed by a comma. The last parameter of a list input via the teletype keyboard is followed by a carriage return. While typing a command language operator, if an error is made, the entire line can be deleted by typing an up arrow (↑) character, followed by a carriage return.

Complete descriptions and details of all command language operators may be found in following sections. However, one special operator, PC, will be discussed here. Process Command (PC) operator terminates a command language operator input stream, after which each of the previously input operators are processed in the input sequence. It should be noted that no operators are processed until a PC operator is received.

Since many of the parameters required by command language operators are Logical Device Numbers (LDN) or device mnemonics, users should become familiar with the assignment of LDN's and mnemonics for the system peripheral devices. Figure 4-1 provides details of these assignments.

Errors encountered while processing command language operators cause prompting messages to be printed on the console teletype. Each type of error recognized produces a unique error message. Those errors requiring an operator decision or intervention cause the general error routine to be entered. Upon entrance, this routine prints the message

CCPE

which informs the operators that a Control Command Processor Error has occurred, and a response must be received before processing may continue. The operator must respond with either a

C, or
X

character, followed by a carriage return.

A response with character C requests that the command in error be ignored and that processing shall continue.

A response with character X requests that further processing be aborted and that MPCS be initialized as is normally done for an end of job.

Following is a list of prompting messages printed for errors detected while processing command language operators.

STANDARD DEVICE LDN AND MNEMONIC ASSIGNMENT

MNEMONIC	DEVICE	STANDARD OCTAL LDN
NO	No Device = Buffer Operations Only	0
KB	ASR 33 Teletype Keyboard	1
TP	ASR 33 Paper Tape Punch	2
TR	ASR 33 Paper Tape Reader	2
PP	High Speed Paper Tape Punch	3
PR	High Speed Paper Tape Reader	3
CP	Card Punch	4
CR	Card Reader	4
LP	Line Printer	5
M0	Magnetic Tape Drive 0	6
M1	Magnetic Tape Drive 1	7
SY	Disk System Area	10
DC	Disk Scratch Area	11
BO	Disk Binary Object	12
M2	Magnetic Tape Drive 2	13
M3	Magnetic Tape Drive 3	14
DO	Diagnostic Output	16
LI	Disk Library Area	17

Figure 4-1

OCTAL ERROR - This message will occur when a command specification numeric value is not expressed as an octal value. Control will then be given to the general error routine for operator response.

OPTION ERROR - This message will occur when a non-valid device mnemonic or specification mnemonic is entered. Control will then be given to the general error routine for operator response.

INVALID - This message will occur when the command is not valid.

NAME ERROR - This message will occur when an external subroutine name is defined with more than four characters. Control is then transferred to the general error routine for operator response.

INVALID ADDR - This message will occur when an address specification is not greater than the resident monitor.

4.1 Command Language Operators

The following narrative of this section provides details for each MPCS command language operator. Figure 4-2 provides a summary and should be used as a quick reference for the operator mnemonics and general parameter format. Section 4.2 provides details of the various display console functions. Explanations of the available operator actions and inputs will be found in that section.

4.1.1 Process Command (PC)

The PC operator terminates command input, initializes the system to accept a new job, and executes previous command inputs.

```
Example 1:   PP  M1
             PC

        2:   FA  BO,CP
             FA  CR,M1
             AS
             PC
```

Example 1 illustrates the PC operator use after a single operator job stream. Example 2 illustrates a PC operator in a typical

COMMAND LANGUAGE OPERATOR SUMMARY

PAGE	MNEMONIC	FUNCTION	GENERAL FORM
34	AF	Advance Magnetic Tape File	AF MAG TAPE LDN OR MNEMONIC
33	AR	Advance Magnetic Tape Record	AR MAG TAPE LDN OR MNEMONIC
22	AS	Assembler Call	AS SP, NL, NO, CP, NS, XX
32	BF	Backspace File	BF MAG TAPE LDN OR MNEMONIC
33	BR	Backspace Magnetic Tape Record	BR MAG TAPE LDN OR MNEMONIC
31	CL	Copy Library	CL NJ
21	CP	Call Program	CP NAME
27	DA	Dump Absolute	DA DEVICE, START, END, ENTRY
20	DC	Device Change	DC DEVICE
21	DN	Delete Name	DN NAME
20	DP	Dump Program	DP START, END, NAME
28	EX	Execute Program	EX LOCATION
30	FA	File Assignment	FA FROM DEVICE, TO DEVICE
24	FC	FORTRAN Call	FC SS, NL, NO, LN, TE, PM, NE, LC, DA
21	FP	Find Program	FP NAME
31	IN	Interrupt Subroutine Names	IN NAME, NAME, ... NAME
34	JT	Generate Job Task Table (JTT)	JT
21	LP	List Program	LP
20	OC	Omit Command	OC
28	OD	Octal Dump	OD START, END
22	OP	Octal Patch	OP LOCATION, PATCH
18	PC	Process Commands	PC
32	PP	Position Peripheral	PP DEVICE
25	RL	Relocating Loader Call	RL LOCATION, MAP DEVICE, NG
31	RS	Restore Standard Assignments	RS
33	SF	Search Magnetic Tape File	SF MAG TAPE, NAME
33	SR	Search Magnetic Tape Record	SR MAG TAPE, NAME
34	SS	Select Peripheral Switch	SS
28	UD	Update Call	UD IN DEVICE, OUT DEVICE, COM DEVICE

Figure 4-2

job stream.

4.1.2 Omit Command (OC)

The OC operator deletes the previous command operator from the input stream.

Example:
```
UDM1, MO, CR
OC
UD M1, MO, CR
```

This example illustrates that OC removes the prior command which contains an error.

4.1.3 Device Change (DC)

The DC operator changes the command input device to the device specified. This operator has the general form:

```
DC XX
```

where: XX equals a two character device mnemonic (see Figure 4-1).

Example:
```
DC CR
```

This example illustrates that the command device is being changed to the card reader. Subsequent command operators will be input from that device. The last command prior to the PC of the job stream input from the card reader should be a DC operator to change the command device back to the keyboard.

4.1.4 Dump Program (DP)

The DP operator dumps the user program in absolute form from memory to the system library and enters the program name and size into the system directory. This operator has the general form:

```
DP XXXXX,YYYYY,NAME
```

where:
- NAME = Symbolic name of user program and may not exceed four characters.
- XXXXX = Starting address of program in core.
- YYYYY = Ending address of program in core.

The standard or default device for this operator is:

Output device LDN '17 LI

Example: DP 40000,40700,TEST

4.1.5 Call Program (CP)

The CP operator calls or loads the program from the system library but does not cause execution of the program. The general form of this operator is:

CP NAME

where: NAME = Symbolic name of users' catalogued program.

Example: CP TEST

4.1.6 Find Program (FP)

The FP operator finds the user program in the system directory and sets the disk control word to point to the disk location where the program resides. The general form of this operator is:

FP NAME

where: NAME = Symbolic name of users' catalogued program.

Example: FP TEST

4.1.7 Delete Name (DN)

The DN operator deletes the user program from the system library and updates the system directory. The general form of this operator is:

DN NAME

where: NAME = Symbolic name of users' catalogued program.

Example: DN TEST

4.1.8 List Program (LP)

The LP operator lists the user programs and processors that

are active on the system library by their designated names. The standard or default device for this operator is:

Listing device LDN 1 KB

Example:
```
LP
PC
```

This example illustrates the LP operator which requests a listing of the current disk directory. A typical listing of a disk directory is shown below:

```
DISK DIRECTORY
JT
LK
FC
AS
UD
OD
DA
RL
LB
```

4.1.9 Octal Patch (OP)

The OP operator enables users to modify any location in memory except those areas which are protected by MPCS. The general form of the operator is:

OP XXXXX,PPPPPPPP

where: XXXXX = Core address to be modified

PPPPPPPP = 8 character octal patch to be placed in location XXXXX.

Example: OP 40000,77777777

4.1.10 Assembler Call (AS)

The AS operator loads the assembly language processor from the system library into memory, adjusts the assembler control word as designated by the optional parameters, and initiates execution of the assembler. This command operator has the general form:

```
AS   SP, NL, NO, CP, NS, XX
```

where:

SP = Causes the assembler to execute in single pass mode. Default for this option is the two pass mode.

NL = Suppresses the symbolic listing output. Default for this option is a symbolic listing of the assembled program.

NO = Suppresses object code input for the assembly. Default for this option is the output of generated object code.

CP = Directs the assembler to generate channel I/O. Default for this option is the generation of unit I/O.

NS = Suppresses the listing of the symbol table. Default for this option is that the symbol table will be output to the listing device.

XX = An optional two character device mnemonic that specifies the secondary second pass symbolic source input device. Default for this option is that the standard assigned device is used for secondary input.

If no optional parameters are required, this operator takes the form:

```
AS
```

This form designates that standard device assignments and all default conditions be in effect for execution of the assembler.

Standard device assignments are as follows:

Source input device	LDN'4	CR
Source listing device	LDN'5	LP
Object output device	LDN'12	BO

Example 1:

```
AS  SP, NL, NO, CP, NS, M0
```

2:

```
FA  BO, CP
FA  CR, M1
AS
```

Example 1 illustrates the AS operator requesting all optional parameters. M0 or magnetic tape unit 0 has been designated as the secondary device. Example 2 illustrates that FA operators may be used to change the standard device assignments. In this example, the source input would be from magnetic tape unit 1 and object output to the card punch.

4.1.11 FORTRAN Call (FC)

The FC operator loads the system FORTRAN compiler from the system library into memory, adjusts the compiler control word as designated by optional parameters, and initiates execution of the compiler. The general form of this operator is:

FC SS, NL, NO, LS, TE, PM, NE, LC, DA

where:

SS = Suppresses printing of sequence numbers if the source output device is the ASR keyboard. Default for this option is that the sequence numbers will be output if a source listing is output.

NL = Suppresses the source listing output. Default for this option is an output of a source listing.

NO = Suppresses the object code output. Default for this option is an output of the object code.

LS = Lists the symbolic assembler code for the compilation. Default for this option is no listing of symbolic assembler code.

TE = Includes the FORTRAN trace feature with the users program. Default for this option is that no trace will be performed.

PM = Indicates that mapping is to be performed. Default for this option is no mapping.

NE = Causes the compiler to generate object code for a system that does not access the EXTENDED ARITHMETIC UNIT (EAU). Default for this option is the generation of EAU instructions in the object code.

LC = Causes FORTRAN to accomplish a special library compilation. Default for this option is that no special library compilation is performed.

DA = Causes FORTRAN to accomplish dynamic addressing. Default for this option is that no dynamic addressing is performed.

If no optional parameters are required, this operator takes the form:

FC

This form designates that both the standard device assignment and the default conditions be in effect for the execution of the compiler.

Standard device assignments are as follows:

Source input device	LDN'4	CR
Source listing device	LDN'5	LP
Object output device	LDN'12	BO
Secondary symbolic input device	LDN'11	DC

Examples:
```
FC NL, NO
FA BO, CP
FC
```

4.1.12 Relocating Loader Call (RL)

The RL operator loads the system relocating loader from the system library into memory, adjusts the loader control word as designated by the command options, and initiates execution of the loader. This operator has the general form:

RL XXXXX, XX, NG

where:

XXXXX = An octal number which specifies the starting address to begin loading the users program. Default for this option is the standard system bias address (20000 octal).

XX = A two character device mnemonic to which a memory map will be output. Default for this option is that a memory map will be output to LDN'5 LP.

NG = Specifies that the loaded program is not to be executed. Default for this option is that the program will be entered for execution if an entrance address was specified.

If no optional parameters are requested, this operator takes the form:

RL

Standard device assignments are as follows:

Primary input device	LDN'12	BO
Secondary input device	LDN'17	LI

The following messages may be printed at the completion of a load. The first two messages are normal, whereas the remainder are error messages and cause an abort of the loader.

MESSAGE	EXPLANATION
DONE	Load completed and program entered for execution.
NOEX	Start address not encountered, user must use "EX" command.
LIBE	Unsatisfied references at end of load.
CKSM	Checksum error in binary record.
SEQN	Sequence error in binary input.
PROTECTED	Trying to load within protected area of core.
SUBR	Subroutine call table full.
SIZE	Incompatible common block sizes.

The MPCS/1 relocating loader is a special adaptation of the SEL 840A relocating loader. The loader will execute in two modes, NORMAL or REDUNDANT. The normal mode is consistent with the original loader except for alterations that are incorporated to allow compatibility with MPCS/1. The present SEL 840A library is not re-entrant as is necessary in a priority interrupt system. Tasks and routines connected to interrupts utilizing a common library subroutine must have the subroutine loaded separately for each task or routine requiring it. This is the REDUNDANT loading mode. Application programmers must take care to insure that jobs structured for the

MPCS/1 environment request redundant loading for all tasks that for any reason may be suspended voluntarily or involuntarily. Suspensions can be invoked because of scheduling higher priority tasks, either explicitly or by the periodic time queue. The redundant mode is invoked by presenting an "IN" (interrupt name) command (see Section 4.1.18) to the system prior to calling the loader. In this mode the loader searches the primary input device for subroutines whose names match those contained in the "IN" command. When a match is made, the input device is switched from primary to secondary (library) and all library references by the main line program are satisfied. When the file end is encountered for the secondary device, the input device is again switched from secondary to primary, the secondary device is rewound, and loading continues until another "IN" name is found to invoke a library search for unsatisfied references of the previously loaded subroutines. The process is repeated until all references are satisfied or an end of file is encountered for the primary device. If a memory map has been requested, it will be output at the end of each redundant load.

Examples:
```
RL
RL 40000
RL 50000,KB
```

4.1.13 Dump Absolute (DA)

The DA operator loads the absolute dump program from the system library into memory and enters it for execution. The general form of this operator is:

```
DA XX,XXXXX,YYYYY,ZZZZZ
```

where:

XX = A two character device mnemonic that specifies the output (dumping) device.

XXXXX = An octal address specifying the starting address of memory to be dumped.

YYYYY = An octal address specifying the last address of memory to be dumped.

ZZZZZ = An octal address specifying the starting address of the program that is to be dumped.

Example:
```
DA CP,40000,42563,40001
```

4.1.14 Octal Dump (OD)

The OD operator loads the octal dump from the system library into memory and initiates its execution. The general form of this operator is:

OD XXXXX,YYYYY

where: XXXXX = An octal address specifying the starting address of memory to be dumped.

YYYYY = An octal address specifying the last address of memory to be dumped.

Example: OD 00275,00500

4.1.15 Execute Program (EX)

This operator directs MPCS to initiate execution of the program previously loaded by the relocating loader or the CP operator. The general form for this operator is:

EX XXXXX

where: XXXXX = An optional octal address specifying the entry location of the user program. If an address is not specified, the last program loaded will be entered for execution.

Examples: EX
EX 40000

4.1.16 Source Update (UD)

The UD operator loads the source update program from the system library and initiates execution of it. The general form of this operation is:

UD XX_1, XX_2, XX_3

where: XX_1 = Symbolic input device

XX_2 = Symbolic output device

XX_3 = Symbolic update command input device

If the standard devices are required, the general form of this operator becomes:

```
UD
```

Standard device assignments are as follows:

Input device	LDN'6	M0
Output device	LDN'7	M1
Command device	LDN'1	KB

Example 1: `UD M1,LP`

2: `UD M0,M1,CR`

Example 1 requests an update from magnetic tape unit 1 to the line printer. Example 2 requests an update from magnetic tape unit 0 to magnetic tape unit 1, with the commands input from the card reader.

Update will recognize and process the following ten, one-character commands:

(1) L = Position output device. Peripheral positioning is dependent upon the I/O Driver.

(2) S = Sequence the output card images. Sequence numbers will appear in card image columns 78-80.

(3) R = Reset input and output card image sequence numbers. The sequence number corresponds to program listing line numbers.

(4) I = Insert card images following this command. Card images are input from command device and written on the output device. Card images will be input until a slash (/) is detected in column 1 of a card image.

(5) / = Terminate insertion mode. Character must appear in column 1 of a card image.

(6) CXXXX = Copy to and including input card image with the sequence number XXXX; where, XXXX is a four character decimal number.

C END = Copy to and including input card image with the characters "END" appearing in columns 5-8.

C EOF = Copy to and including magnetic tape end-of-file.

(7) DXXXX = Delete input card images to and including card image XXXX; where, XXXX is a four-character decimal number.

D END = Delete input card images to and including the card image with the characters "END" appearing in columns 5-8.

D EOF = Delete input card images to and including magnetic tape end-of-file.

(8) F = Write end-of-file on output magnetic tape transport.

(9) O = Omit last command input.

(10) E = Process commands previously input. When processing is complete, exit is made to MPCS.

4.1.17 File Assignment (FA)

The FA operator enables a user to assign a Logical Device Number (LDN) to a symbolic device or to reassign standard devices. The general form of this operator is:

FA XXX_1,XX_2

where:

XXX_1 = The current LDN (3 octal characters) or a two character mnemonic (see Figure 4-1). This parameter may also be a user defined LDN which may be any octal number from 000 to 777 but must consist of three octal characters.

XX_2 = A two character mnemonic which designates the new device assignment (see Figure 4-1).

Examples:
```
FA BO,CP
FA CR,M1
FA 012,CP
FA 004,M1
```

4.1.18 Interrupt Subroutine Names (IN)

This operator directs MPCS to store in memory the names of user defined subroutines that are to be connected to external interrupts during execution of the job. The list of names is processed by the Relocating Loader (see Section 4.1.12). The general form of this operator is:

IN $XXXX_1, XXXX_2, \ldots XXXX_N$

where: $XXXX_1$-$XXXX_N$ = 1-4 character names and the maximum number of names (N) is 15.

Example: IN $TES_1, TES_2, ROUT, INT_2$

4.1.19 Restore Standard Assignments (RS)

The operator restores the system devices to their standard assignment and sets the magnetic tape mode to BCD. See Figure 4-1 for the standard device assignments.

Example:
```
RS
UD M0,M1,CR
```

This example illustrates the RS operator used prior to an update request.

4.1.20 Copy Library (CL)

This operator directs MPCS to copy object modules. Binary modules to be copied must be in the standard SEL format produced by the assembler or FORTRAN compiler. Checks are made to verify that the modules input are correct. The general form for this operator is:

CL NJ

where: NJ = An optional parameter which when present specifies that end-of-job object modules are not to be copied to the output device. When this parameter is not present, end-of-job modules will be copied.

Standard device assignments for this operator are as follows:

Input device	LDN'17	LI
Output device	LDN'12	BO

Example:
```
CL NJ
CL
```

4.1.21 Position Peripheral (PP)

The PP operator positions or initializes the specified peripheral device. The general form of this operator is:

```
PP XX
```

where: XX = A two character device mnemonic (see Figure 4-1).

The following actions are performed for the peripheral devices:

Magnetic Tape	-	Rewind to load point
Disk	-	Initialized to scratch area
Paper tape punch	-	Paper tape leader is punched
Line Printer	-	Slew to top of form
Keyboard	-	Skip two lines
Card reader	-	No action
Card punch	-	No action

Example:
```
PP M0
PP M1
```

This example illustrates a request to rewind magnetic tape units 1 and 0.

4.1.22 Backspace File (BF)

The BF operator directs MPCS to backspace the specified magnetic tape drive one file. The general form of this operator is:

```
BF XX
```

where: XX = A two character magnetic tape device mnemonic.

Example:
```
BF M0
```

4.1.23 Search Magnetic Tape Record (SR)

The SR operator directs MPCS to search the specified magnetic tape until a record is read that contains, as the first four characters, a name equal to the name specified in the command. The general form of this operator is:

SR XX, AAAA

where: XX = A two character magnetic tape device mnemonic.

AAAA = A four character record name that is to be found.

Example: SR M1, SPID

4.1.24 Search Magnetic Tape File (SF)

The SF operator directs the MPCS to search the specified magnetic tape until a file is found whose first record contains the specified name as its first four characters. The tape is then backspaced one record. The general form of this operator is:

SF XX, BBBB

4.1.25 Backspace Magnetic Tape Record (BR)

The BR operator directs MPCS to backspace the specified magnetic tape one record. The general form of this operator is:

BR XX

where: XX = A two character magnetic tape device mnemonic.

Example: BR M2

4.1.26 Advance Magnetic Tape Record (AR)

The AR operator directs MPCS to advance the specified magnetic tape one record. The general format of this operator is:

AR XX

where: XX = A two character magnetic tape device mnemonic.

Example: AR M0

4.1.27 Advance Magnetic Tape File (AF)

The AF operator directs MPCS to advance the specified magnetic tape one file. The general form of this operator is:

```
AF  XX
```

where: XX = A two character magnetic tape device mnemonic.

Example: AF M1

4.1.28 Select Peripheral Switch (SS)

The SS operator directs MPCS to connect the peripheral switch to the processor making the request. The general form of this operator is:

```
SS
```

Example: SS

4.1.29 Generate Job Task Table (JT)

The JT operator causes MPCS to load and execute the JTT generation program (see Section 4). The standard devices for this program are:

Output device	LDN '12	BO
Listing device	LDN '5	LP

The general form of this operator is:

```
JT
```

Example:
```
JT
AS
RL
EX
```

4.2 Display Console Functions

Applications designed to utilize the MPCS display services will ordinarily involve a certain amount of activity on the part of the display console operator. The amount depends on how the application is organized.

Following the initialization of a display-oriented application as described in Section 2.3, the operator is capable of interacting with the application tasks and displays through the use of three basic devices.

(1) Display console keyboard

(2) Light pen

(3) Function switches

The first two are used in conjunction with predefined fields in the displays used by an application.

Normal system operation will progress in response to the actions of the display console operator and the requests from the application tasks. In the event that error situations are encountered, error messages will be displayed in the one-line message area at the bottom of the screen. These error messages are summarized in Section 6.3.

4.2.1 Keyboard Input

For displays containing compose fields defined by one or more slash (/) characters, the operator can perform any of the following:

(1) Position the data cursor to any compose field character.

(2) Enter a keyboard character into the character field to which the cursor is currently positioned.

(3) Cause the contents of the compose field currently selected by the cursor to be transmitted to the 840MP for processing.

The use and significance of the keyboard characters are illustrated in Figure 4-3.

Cursor control is provided by five keys which move the cursor left, right, up, down, and home (to the beginning of the first compose field). Cursor left will move the cursor to the preceding character position in the current compose field. Wrap-around is provided so that moving the cursor left from the first character position will cause it to be moved to the last character of the current compose field.

DISPLAY CONSOLE CHARACTER SET

Group 1	ALPHABETIC	A - Z
	NUMERIC	0 - 9
	SPECIAL	" # ' ( ) * = - + , . ? / 〈 〉 space
Group 2	SPECIAL	! $ % & ← ↑ [ ] \
Group 3	SPECIAL	; : @ RETURN, LINE FEED, DELETE

Process / Group	Output to Typewriter	Displayed on CRT	Stored in Input Buffer	*Cursor Control	Ignored
1	x	x	x		
2	x				x
3	x			x	

* Cursor control functions are as follows:

KEY	FUNCTION
;	Cursor right
Delete	Cursor left
:	Cursor up
Line feed	Cursor down
@	Cursor home (to first compose field)
Return	Transmit data to task

Figure 4-3

Cursor right will move the cursor to the next character of the current compose field with wrap-around from last-to-first. Cursor up and cursor down move the cursor to the first character position of the preceding and next compose fields, respectively. Wrap-around is provided for first-to-last (up) and last-to-first (down). It should be noted that the cursor left and right functions do not affect data previously entered into the input buffer but that cursor up, down, and home always reset the input buffer with blanks.

As each data character is entered it is output to the typewriter and, if it is a valid data character, displayed on the screen and inserted into the input data buffer. If it is not a valid data character, it will be ignored. After a valid input character has been processed, the cursor is moved automatically to the next character position unless it is currently positioned at the last character of a compose field. In such a case it will not be moved and subsequent input characters will be entered into the last character of the compose field unless the cursor is manually repositioned by the operator.

After the desired data has been entered into a given compose field, depressing the RETURN key will cause the data to be transmitted to the 840MP for processing by an application task. This action may also cause a new display to appear on the screen, depending on whether or not a new display was assigned to the compose field at library generation time.

4.2.2 Light Pen Input

One or more light pen options will exist for all displays. They are identifiable by the less than (<) sign and greater than (>) sign which begin and end them respectively. Touching the light pen beam to an option and depressing the button on the barrel of the light pen results in actions similar to those caused by the use of the RETURN key for a compose field. A new display may or may not appear on the screen and a task may or may not be scheduled in the 840MP. Again, the results depend on the manner in which the option was defined at library generation time.

Two standard pen options appear at the bottom of most displays. The RETURN TO PRIOR LEVEL option, when selected, will cause the immediately preceding display to replace the current display. No task gets scheduled in response to this option. This option appears on all displays and, in fact, is the only option allowed in special displays. The second standard option is that for the one-line message facility. It is designed for use by application tasks and, unless it has

a new display and/or task assigned to it by an application task, it will provide no response when selected.

Light pen interrupts are not processed for a period of approximately one second following the loading of a new display. This is to allow the operator sufficient time to remove the light pen to avoid inadvertent selection of a pen option on the new display.

4.2.3 Function Switch Input

While operator use of the keyboard and light pen depends on the format of the predefined display appearing on the screen at any given time, use of the display console function switches is display independent. A function switch can be used to activate a task regardless of which display is active. However, in order for a function switch to be used in such a fashion, it must have a task attached to it. System services are provided to allow application tasks to be attached to or detached from the function switches. Once a task is attached to a switch, the task gets scheduled in the 840MP each time the operator raises the switch. The function switches on the left side of the screen are numbered 1 through 8 (top-to-bottom) and those on the right, 9 through 16.

5. DISPLAY LIBRARY GENERATION

"MPCS/1 Users' Guide, Section 3, Volume 1, Programmers' Guide", provides details of the Display Library Language (DLL) and methods of defining the specifications for a particular display. This section will provide details for generating the display library which, when finalized, resides in the 810B disk. This procedure requires two distinct steps:

(1) Process card images of the DLL.

(2) 810B disk library generation which processes the tape produced by the previous step above.

5.1 Process DLL Card Images

The Display Librarian Program is resident on the SEL 840MP disk and is loaded and executed by the following command language sequence:

```
CP  DL
EX
PC
```

Resources required by the Display Librarian Program include the:

- Card Reader
- Line Printer
- One Magnetic Drive (LDN 6, Unit 0)

A second magnetic tape drive, LDN 13, unit 2, is required if the DLL source is contained on magnetic tape. Figure 5-1 provides the required deck setup.

For initial creation of the display library, a magnetic tape is mounted on drive 0, the input data cards for display 0001 (and for all other displays to be included in the display library at this time) are stacked in the card reader, and the Display Librarian Program is called for execution. When execution of the Display Librarian Program has completed, the magnetic tape mounted on drive 0 will contain the display library.

For subsequent display library updates, the previously created and/or updated display library magnetic tape is mounted on drive 0, the input data cards for the new display or displays are stacked in the

DISPLAY LIBRARIAN INPUT DECK SETUP

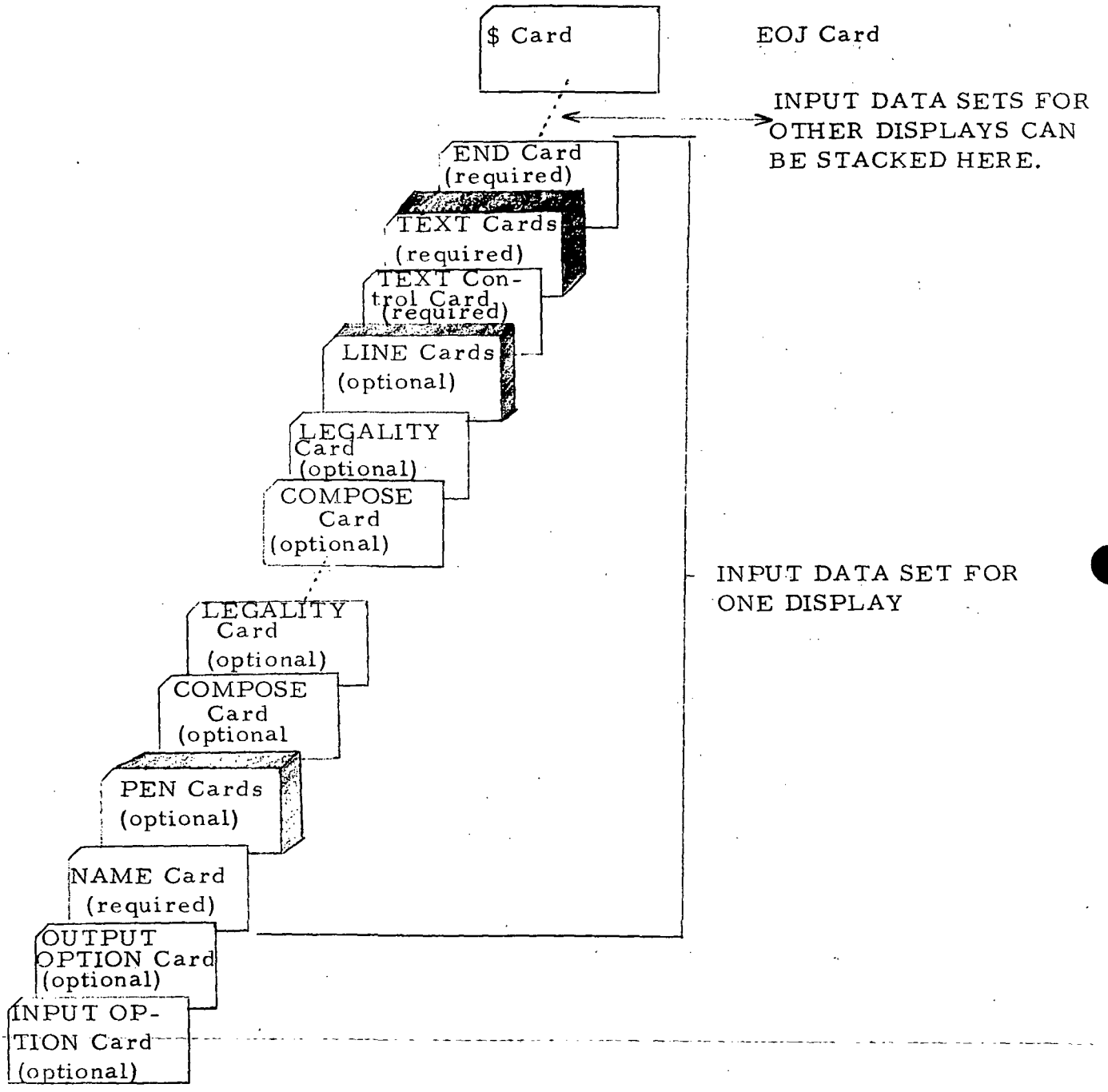


Figure 5-1

card reader, and the Display Librarian Program is called for execution. This procedure will add the new display or displays to the Display Library magnetic tape.

5.2 810B Disk Library Generation

Formatted displays on the magnetic tape produced by the Display Librarian must be loaded into the 810B online disk library before they can be utilized by the MPCS Display Processor. This process involves the simultaneous execution of the 840MP Display Library Loader and the 810B Display Library Loader programs. The former reads the displays from magnetic tape and transmits them to the 810B. The latter program receives the displays from the 840MP and stores them in 810B auxiliary (disk) storage.

Steps for loading the display library are as follows:

(1) Initialize the 840MP and 810B systems as discussed in Section 2.

(2) Load and activate the 810B Display Library Loader by entering the following commands via the 810B console keyboard.

```
/POSF, BI, MPCSLL.
/XFER, BI, F=0, L=3000.
/GOTO, R=300.
```

At this point the program will issue the following message to the typewriter.

```
LIBRARY LOADER READY
```

(3) Mount the magnetic tape on logical tape unit M0 of the SEL 840MP.

(4) Load and activate the 840MP Display Library Loader by entering the following commands via the 840MP command device:

```
CP LL
EX
PC
```

After all displays on the tape have been loaded, the

following message will be printed on the 810B typewriter:

LIBRARY COMPLETED

A listing of display numbers (tags) will be output on the 840MP printer.

6. SYSTEM ERROR SUMMARY

6.1 Error Conditions Causing User Abort

ERROR MESSAGE	EXPLANATION
STALL ERR	Stall alarm interrupt became active (instruction counter not incremented after 32 cycles).
INST TRP	Application program attempted to execute one of the following instructions: PON AOP TAC CEU PIE MOP PIR POF AIP TEU LBR PID MIP
HLT ABRT	Operator requested an abort of an application program executing a halt instruction (HLT).
NO MSERV	Application program requested MPCS service not currently assigned.
MEM PROT	Application program attempted to branch to or store into a protected area.
STK OVFL	MPCS interrupt save stack overflowed due to excessive imbedded interrupts becoming active.
LDN ERR	A request for I/O was made to MPCS and the Logical Device Number was not a standard LDN and was not assigned via an "FA" command.
I/O ERR	A non-recoverable I/O ERROR was encountered by MPCS and the operator requested an abort.
CNSL ABN	Application program was aborted by an operator causing a console interrupt.
USER ABN	Application program requested the abnormal end of job MPCS service.

ERROR MESSAGE	EXPLANATION
HALT	User program has attempted to halt the CPU via a halt instruction. An operator response is required where: C causes a return to program and a continued execution. X causes an abort of the program.

6.2 I/O Error Conditions for General I/O Devices

The general form for the I/O error message is:

```
CCCC
LLL
```

where: CCCC is a four character error message.

LLL is a three character LDN of the device causing the error.

An operator response is required, where the response character is followed by a carriage return:

C = Continue operation and ignore error
R = Retry operation causing error
X = Abort program, return to MPCS

An X response causes the message

```
I/O ERR
```

to be printed.

The following table (Table 6-1) defines the general I/O error messages and an explanation of each.

6.3 Display Message Summary

MPCS-DP0	Display processor is ready, awaiting 840 request.
MPCS-DP1	Requested task does not exist, no TQI could be found.

I/O ERROR CONDITIONS

MESSAGE	EXPLANATION
MTRE	Magnetic Tape Read Error
EOFR	EOF while reading
MTWE	Magnetic Tape Write Error
EOT	End of Tape
EOFE	EOF Error
PLOW	Paper Low on Line Printer
DCKE	Disk Checksum Error
DFUL	Disk is Full
CRER	Card Reader Error
CPER	Card Punch Error
OUFE	Data Underflow or Overflow
HANG	Device is not ready or off line

Examples:

1.
```
EOFR
 006
```

2.
```
PLOW
 005
C
```

Example 1 illustrates an End of File error while reading tape unit 0 (LDN 6) or M0. Example 2 illustrates a paper low condition for LDN 5 or LP. The character C illustrates an operator response, which in this case requests that MPCS continue at the point where the error occurred.

Table 6-1

MPCS-DP2	Requested task is currently executing.
MPCS-DP3	Task communication area is too small for compose field.
MPCS-DP4	Requested task has not yet processed previous request.
MPCS-DP5	Requested task has no facilities for accepting input data.
MPCS-DP6	Simultaneous function switch settings not accepted, retry.
MPCS-DP7	Function switch currently has no task attached to it.
MPCS-DP8	An error has been detected in an 810/840 transmission.
MPCS-DP9	There are no compose fields in this display.
MPCS-DP10	A pen interrupt was detected but coordinates were invalid.
MPCS-DP11	Display requested from 840, select option to activate.
MPCS-DP12	Prior level request is invalid, this is initial display.
MPCS-DP13	Requested display could not be found in the display library.
MPCS-DP14	Data entered is out of limits for this subfield, re-enter.
MPCS-DP15	No response received from 840.
MPCS-DP16	Character type is invalid for this subfield, re-enter.
MPCS-DP17	Halt display library has been destroyed.
MPCS-DP18	Halt invalid 840 request.

6.4 I/O Error for Hybrid Services

All hybrid I/O services provide error checking unless the user declares in his program a memory location with the symbolic name "NERR" and sets it to -1. If "NERR" is not declared by the user, a word is loaded from the system library that has the value of +1 to indicate that error checking will be performed. Hybrid I/O service routines that encounter an error will call H$ER (system library routine) to print a message indicating the name of the routine which encountered an error and the type of error before returning to the user's calling sequence.

7. SYSTEM GENERATION

In order to create the Multi-Processor Control System (MPCS/1), certain steps must be followed. This section will describe these steps and explain the procedures used to execute them. For all steps the Peripheral Switch must be in a local position (1, 2, or 3).

7.1 Bootstrap Deck Creation

The first step in generating the system is to obtain an absolute "System Library Generator" card deck. This deck is obtained as follows:

(1) First, using the "SEL 840A/MP Assembly Program", process the "System Library Generator" symbolic instruction input (source input) to create an object program output.

(2) Next, using any "SEL 840A/MP Relocating Loader Program", load the "System Library Generator" object program into core locations that will not be used by any of the standard system programs. The "System Library Generator" is relocatable and requires less than 2100 decimal locations. At the present time it is being loaded into a BSS area of MPCS/1.

(3) After the "System Library Generator" object program is loaded, use the "SEL 840A/MP Relocating Loader Program" to load the "Card Dump For Hardware Loader" program. This program may be loaded into any core location except where the "System Library Generator" was loaded. The "Card Dump For Hardware Loader" program is then used to generate the desired absolute bootstrap card deck of the "System Library Generator" as follows:

As the "Card Dump For Hardware Loader" is entered the message "MP or A" will be typed. The user's response is "MP" followed by a carriage return.

Next, the message "LOW" will be typed. The response to this message is the low address of the "System Library Generator" that was printed by the relocating loader when the "System Library Generator" was loaded into core.

The last message that will be typed is "HIGH". The response to this message is the high address of the "System Library Generator" that was printed by the relocating loader.

After the response to "HIGH" is made, the card deck will be output.

7.2 System Tape Creation

The next step is to create a system tape and disk from system program object decks. This is accomplished as follows:

(1) Mount a scratch tape on logical tape drive M0.

(2) Next, place the absolute bootstrap deck described in Section 7.1 and the system program object decks in the card reader. It should be noted that a Title Card, Figure 7-1, must precede each object deck and an "END" Card must be the last card in the last deck. See Figure 7-2 for deck setup example. The decks may be in any order with the following exception: the bootstrap deck must be first and the END card must be last.

(3) When the "Load Switch" on the console is depressed, the cards will be read into memory, linked, and written on logical tape drive M0. At this point "END" is typed on the console typewriter, logical tape M0 is rewound, and the central processor halts.

This completes the creation of the System Tape. The next step is to create the System Disk.

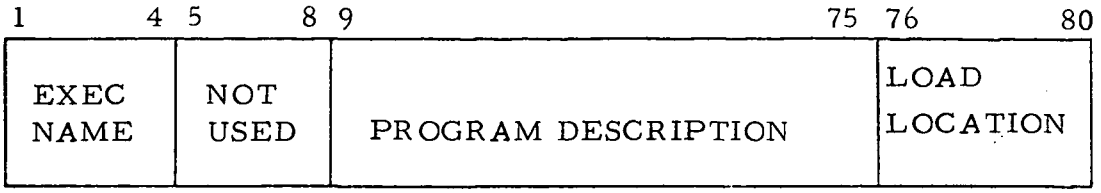
7.3 System Disk Creation

The System Disk is created from the System Tape that was described in Section 7.2. This may be accomplished by either of the following procedures.

7.3.1 Complete Initialization Procedure

First is the procedure where the system tape has just been created and the central processor is halted. A sense switch option, sense switch zero, is available to the user at this point that will

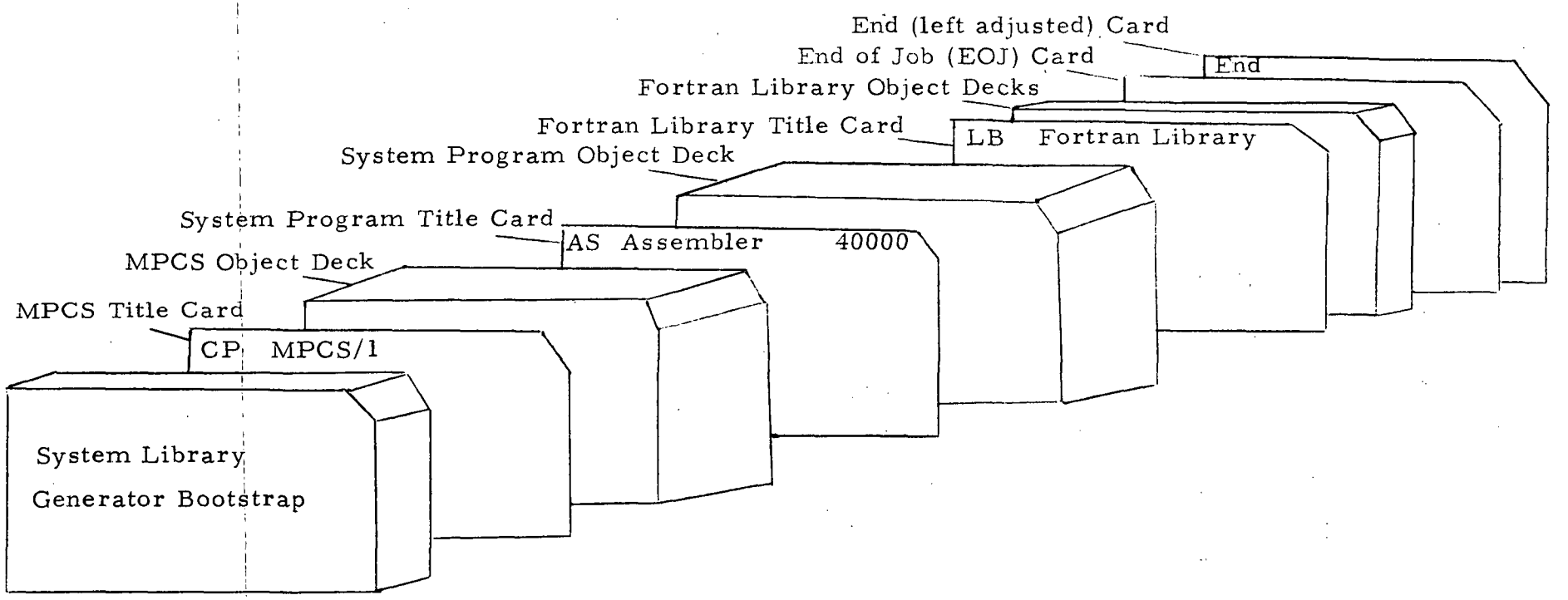
TITLE CARD FORMAT



Columns 1 - 4	The symbolic name, left adjusted, that is used by MPCS to load the program.
Columns 5 - 8	Not used.
Columns 9 - 75	Program Description
Columns 76 - 80	The five character octal location in core that the program will be loaded. This location must not reference BAR 0 or be located in the same area as the "System Library Generator".
Note:	Columns 76 - 80 are not used on the MPCS, CP title card or the Fortran Library, LB, title card.

Figure 7-1

SYSTEM GENERATION DECK SETUP



Note: An "EOJ" card is used only at the end of the Fortran Library Object Decks.

Figure 7-2

cause a "bootstrap card" to be punched when the start switch on the console is depressed. The use of this card will be described later.

When the start switch is depressed, each program on the System Tape is written on the System Disk. As each program is written, the disk directory information is listed on the printer in the format shown below:

DISK LIBRARY PRØGRAMS

LØW LØCATIØN	HIGH LØCATIØN	STARTING LØCATIØN	MPCS NAME	PRØGRAM DESCRIPTIØN
55000	57225	55000	JT	JØB TASK TABLE BUILD PRØGRAM
40000	54463	40000	FC	FØRTRAN CØMPILER FØR SEL 840A/MP
40000	55334	40000	AS	ASSEMBLER FØR SEL 840A/MP
20000	20756	20000	UD	SØURCE UPDATE
77600	77761	77600	ØD	ØCTAL DUMP TØ LINE PRINTER
77500	77704	77500	DA	DUMP FØR RELØCATABLE LØADER
75000	77753	75000	RL	RELØCATING LØADER
			LB	FØRTRAN LIBRARY

After all programs are written on the System Disk, the disk directory is listed on the console typewriter, the System tape is rewound, and control is transferred to MPCS. The format for a typical disk directory is shown below:

DISK DIRECTORY OF ACTIVE PROGRAMS

JT
LK
AS
UD
OD
DA
RL
FC
LB

7.3.2 Partial Initialization Procedure

The second procedure is where the System Tape was created and other jobs have since been loaded into the central processor. In order to create the System Disk, it is necessary to load the "bootstrap card" described in Section 7.3.1, by using the load switch on the console.

The contents of this card will load the "System Library Generator" and cause it to execute in the manner described in Section 7.3.1.

7.4 810B System Generation

Two components of the MPCS display support software operate in the SEL 810B and reside in 810B auxiliary storage:

- o MPCS Display Library Loader
- o MPCS Display Processor

Both exist in the form of SEL 810 Assembly Language source cards and are generated into operational software through application of the following procedures:

(1) Assemble the two source decks to produce object code paper tapes.

(2) Generate the SEL 810A/810B Operating System (OS) in the manner described in the SEL Reference Manual 321-095071-000.

(3) Load the paper tape for the MPCS Display Library Loader into the paper tape reader and enter the following commands via the 810B console keyboard to read the tape into the computer:

```
/ASGN, BI=FT, BO=MD.
/LOAD, R=0, Z=0, U.
```

(4) Dump the program to disk storage using the following typewriter commands:

```
/NAME, BO, MPCSLL.
/DUMP, BO, F=0, L=3000.
```

(5) Load the MPCS Display Processor tape into the paper tape reader and read it into memory with the following command:

```
/LOAD, R=0, Z=0, U.
```

(6) Dump the program to disk storage with the following commands:

```
/NAME, BO, MPCSDP.
/DUMP, BO, F=0, L=5000.
/ASGN, BI=MD.
/DUMP, LB, F=0, L=1.
```

The Display Processor and Library Loader can now be exercised as described in Sections 2.3 and 5.2, respectively.

Note that a display library must be created on the 810B disk storage device before the Display Processor can be utilized (see Section 3).